Docket No.: 109870-130097   **MAIL STOP: APPEAL BRIEF-PATENTS**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

By: _(signature)_   Date: <u>June 12, 2006</u>

    Yvette L. Chriscaden

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

<u>Before the Board of Patent Appeals and Interferences</u>

App. No.	:	10/089,139	Confirmation No.:	2275
Inventor	:	Bosworth et al.		
Filed	:	August 19, 2002		
Title	:	A MULTI-LANGUAGE EXECUTION METHOD		
Art Unit	:	2191		
Examiner	:	Rampuria, Satish		
Customer No.	:	25,943		

MAIL STOP: APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Dear Sir:

This appeal furthers the Notice of Appeal filed on April 10, 2006. The appeal arises from a final decision by the Examiner in the final Office Action, dated January 10, 2006. The final decision was in response to arguments filed on October 18, 2005, in response to an earlier office action, mailed July 27, 2005.

Appellants submit this *Brief on Appeal*, including payment in the amount of $500.00 to cover the fee for filing the *Brief on Appeal*. Appellants respectfully request

consideration of this appeal by the Board of Patent Appeals and Interferences for allowance of the present patent application.

Real Party in Interest:

The Real Party in Interest is BEA Systems, Inc., of 2315 North First Street, San Jose, California 95131, which wholly owns B-1 Acquisition Corp., which is a successor in interest to Crossgain Corporation, assignee of the application. Assignment of the application from the Inventors to Crossgain Corporation is recorded with the United States Patent and Trademark Office on August 19, 2002, at Reel 013198 Frame 0130.

Related Appeals and Interferences:

To the best of Appellants' knowledge, there are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

Status of Claims:

Appellants appeal the rejection of claims 1-38. Claims 1-38 were pending and claims 1-38 were rejected in the final Office Action dated January 10, 2006. Applicants respectfully note that, although claims 14-19 are listed by the Examiner as rejected, no rejection has been provided for those claims in any of the office actions. Claims 1-38 are reproduced, as pending, in Appendix A.

Status of Amendments:

Appellants have offered no amendments subsequent to the Examiner's final Office Action.

Summary of the Claimed Subject Matter:

Independent claim 1 is directed towards a *method of computing* that comprises
> "reading a data processing representation having code sections with code
> statements of at least a first and a second programming language;

-2-

recognizing a first code section with at least code statements of a first
programming language;

invoking a first code statement processing unit of the first programming language
to process the first code section;

recognizing a second code section with at least code statements of a second
programming language; and

invoking a second code statement processing unit of the second programming
language to process the second code section."

Figure 1 illustrates an overview of the operational flow of the invention recited in claim 1.
Figure 1 is described in detail on page 4, line 26 through page 6, line 11, in accordance
with some embodiments. Figures 2a-2b are flowcharts illustrating selected operations
recited in claim 1. Figures 2a-2b are described in detail on page 9, line 15 through page
11, line 6, in accordance with some embodiments. Figure 3 illustrates one example of a
computer system capable of performing the operations recited in claim 1. Figure 3 is
described in detail on page 11, line 26 through page 12, line 11, in accordance with
some embodiments. Figure 4 illustrates an example multi-language data processing
representation of claim 1 (Multi-Language DP Representation 106). Figure 4 is
described in detail on page 6, line 14 through page 9, line 15, in accordance with some
embodiments.

Independent claim 14 is directed towards a *method of computing* that comprises:
"reading a data processing representation having code sections with code
statements of at least a first and a second programming language;

recognizing a header section of a selected one of the first and the second
programming language;

recognizing a directive statement within the header section, enumerating one or
more data packages; and

importing the enumerated one or more data packages for use by code sections
within code sections with at least statements of the selected first and
second programming language."

- 3 -

Figure 1 illustrates an overview of the operational flow of the invention recited in claim 14. Figure 1 is described in detail on page 4, line 26 through page 6, line 11, in accordance with some embodiments. Figures 2a-2b are flowcharts illustrating selected operations recited in claim 14. Figures 2a-2b are described in detail on page 9, line 15 through page 11, line 6, in accordance with some embodiments. Figure 2c is a flowchart illustrating selected operations recited in claim 14. Figure 2c is described in detail on page 11, lines 7-23, in accordance with some embodiments. Figure 3 illustrates one example of a computer system capable of performing the operations recited in claim 14. Figure 3 is described in detail on page 11, line 26 through page 12, line 11, in accordance with some embodiments. Figure 4 illustrates an example multi-language data processing representation of claim 14 (Multi-Language DP Representation 106). Figure 4 is described in detail on page 6, line 14 through page 9, line 15, in accordance with some embodiments.

Independent claim 17 is directed towards *a method of computing* that comprises:

> "reading a data processing representation having code sections with code statements of at least a first and a second programming language;
>
> recognizing a header section of a selected one of the first and the second programming language;
>
> recognizing a first declare statement within the header section, enumerating one or more processing methods; and
>
> instantiating the enumerated one or more processing methods for use within code sections with at least statements of the selected first and second programming language."

Figure 1 illustrates an overview of the operational flow of the invention recited in claim 17. Figure 1 is described in detail on page 4, line 26 through page 6, line 11, in accordance with some embodiments. Figures 2a-2b are flowcharts illustrating selected operations recited in claim 17. Figures 2a-2b are described in detail on page 9, line 15 through page 11, line 6, in accordance with some embodiments. Figure 2c is a flowchart illustrating selected operations recited in claim 17. Figure 2c is described in

detail on page 11, lines 7-23, in accordance with some embodiments. Figure 3 illustrates one example of a computer system capable of performing the operations recited in claim 17. Figure 3 is described in detail on page 11, line 26 through page 12, line 11, in accordance with some embodiments. Figure 4 illustrates an example multi-language data processing representation of claim 17 (Multi-Language DP Representation 106). Figure 4 is described in detail on page 6, line 14 through page 9, line 15, in accordance with some embodiments.

Independent claim 19 is directed towards a *method of computing* that comprises:

> "reading a data processing representation having code sections with code statements of at least a first and a second programming language;
> recognizing a header section of a selected one of the first and the second programming language;
> recognizing a declare statement within the header section, enumerating one or more instance variables; and
> instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language."

Figure 1 illustrates an overview of the operational flow of the invention recited in claim 19. Figure 1 is described in detail on page 4, line 26 through page 6, line 11, in accordance with some embodiments. Figures 2a-2b are flowcharts illustrating selected operations recited in claim 19. Figures 2a-2b are described in detail on page 9, line 15 through page 11, line 6, in accordance with some embodiments. Figure 2c is a flowchart illustrating selected operations recited in claim 19. Figure 2c is described in detail on page 11, lines 7-23, in accordance with some embodiments. Figure 3 illustrates one example of a computer system capable of performing the operations recited in claim 19. Figure 3 is described in detail on page 11, line 26 through page 12, line 11, in accordance with some embodiments. Figure 4 illustrates an example multi-language data processing representation of claim 19 (Multi-Language DP

Representation 106). Figure 4 is described in detail on page 6, line 14 through page 9, line 15, in accordance with some embodiments.

Independent claim 20 is directed towards *an apparatus* that comprises:

" at least one storage unit having stored thereon programming instructions designed to enable the apparatus to

read a data processing representation having code sections with code statements of at least a first and a second programming language,

recognize a first code section with code statements of at least the first programming language,

invoking a first code statement processing unit of the first programming language to process the first code section,

recognize a second code section with code statements of at least the second programming language,

invoking a second code statement processing unit of the second programming language to process the second code section; and

at least one processor coupled to said at least one storage unit to execute said programming instructions."

Figure 1 illustrates an overview of the operational flow of the invention recited in claim 20. Figure 1 is described in detail on page 4, line 26 through page 6, line 11, in accordance with some embodiments. Figures 2a-2b are flowcharts illustrating selected operations of the apparatus recited by claim 20. Figures 2a-2b are described in detail on page 9, line 15 through page 11, line 6, in accordance with some embodiments. Figure 3 illustrates one example of the apparatus claimed by claim 20. Figure 3 is described in detail on page 11, line 26 through page 12, line 11, in accordance with some embodiments. Figure 4 illustrates an example multi-language data processing representation of claim 20 (Multi-Language DP Representation 106). Figure 4 is described in detail on page 6, line 14 through page 9, line 15, in accordance with some embodiments.

Independent claim 33 is directed towards *an apparatus* that comprises:

"at least one storage medium having stored therein a plurality of programming

instructions designed to enable the apparatus to

read a data processing representation having code sections with programming

language statements of at least a first and a second programming

language,

recognize a header section of a selected one of the first and the second

programming language;

recognizing a directive statement within the header section, enumerating one or

more data packages, and

import the enumerated one or more data packages for use code sections with at

least code statements of the selected one of the first and the second

programming language; and

at least one processor coupled to the storage medium to execute the

programming instructions."

Figure 1 illustrates an overview of the operational flow of the invention recited in claim 33. Figure 1 is described in detail on page 4, line 26 through page 6, line 11, in accordance with some embodiments. Figures 2a-2b are flowcharts illustrating selected operations of the apparatus recited by claim 33. Figures 2a-2b are described in detail on page 9, line 15 through page 11, line 6, in accordance with some embodiments. Figure 2c is a flowchart illustrating selected operations of the apparatus recited by claim 33. Figure 2c is described in detail on page 11, lines 7-23, in accordance with some embodiments. Figure 3 illustrates one example of the apparatus claimed by claim 33. Figure 3 is described in detail on page 11, line 26 through page 12, line 11, in accordance with some embodiments. Figure 4 illustrates an example multi-language data processing representation of claim 33 (Multi-Language DP Representation 106). Figure 4 is described in detail on page 6, line 14 through page 9, line 15, in accordance with some embodiments.

Independent claim 36 is directed towards *an apparatus* that comprises:

"at least one storage medium having stored therein a plurality of programming

instructions designed to enable the apparatus to

read a data processing representation having code sections with code

statements of at least a first and a second programming language,

recognize a header section of a selected one of the first and the second

programming language,

recognize a first declare statement within the header section, enumerating one or

more processing methods, and

instantiate the enumerated one or more processing methods for use within code

sections with at least code statements of the selected one of the first and

the second programming language; and

at least one processor coupled to the storage medium to execute the

programming instructions."

Figure 1 illustrates an overview of the operational flow of the invention recited in claim 36. Figure 1 is described in detail on page 4, line 26 through page 6, line 11, in accordance with some embodiments. Figures 2a-2b are flowcharts illustrating selected operations of the apparatus recited by claim 36. Figures 2a-2b are described in detail on page 9, line 15 through page 11, line 6, in accordance with some embodiments. Figure 2c is a flowchart illustrating selected operations of the apparatus recited by claim 36. Figure 2c is described in detail on page 11, lines 7-23, in accordance with some embodiments. Figure 3 illustrates one example of the apparatus claimed by claim 36. Figure 3 is described in detail on page 11, line 26 through page 12, line 11, in accordance with some embodiments. Figure 4 illustrates an example multi-language data processing representation of claim 36 (Multi-Language DP Representation 106). Figure 4 is described in detail on page 6, line 14 through page 9, line 15, in accordance with some embodiments.

Independent claim 38 is directed towards *an apparatus* that comprises:

"at least one storage medium having stored therein a plurality of programming

instructions designed to enable the apparatus to

read a data processing representation having code sections with code
statements of at least a first and a second programming language,

recognize a header section of a selected one of the first and the second
programming language,

recognize a declare statement within the header section, enumerating one or
more instance variables,

instantiate the enumerated one or more instance variables for use within code
sections with at least code statements of the selected one of the first and
the second programming language; and

at least one processor coupled to the storage medium to execute the
programming instructions."

Figure 1 illustrates an overview of the operational flow of the invention recited in claim
38. Figure 1 is described in detail on page 4, line 26 through page 6, line 11, in
accordance with some embodiments. Figures 2a-2b are flowcharts illustrating selected
operations of the apparatus recited by claim 38. Figures 2a-2b are described in detail
on page 9, line 15 through page 11, line 6, in accordance with some embodiments.
Figure 2c is a flowchart illustrating selected operations of the apparatus recited by claim
38. Figure 2c is described in detail on page 11, lines 7-23, in accordance with some
embodiments. Figure 3 illustrates one example of the apparatus claimed by claim 38.
Figure 3 is described in detail on page 11, line 26 through page 12, line 11, in
accordance with some embodiments. Figure 4 illustrates an example multi-language
data processing representation of claim 38 (Multi-Language DP Representation 106).
Figure 4 is described in detail on page 6, line 14 through page 9, line 15, in accordance
with some embodiments.

Grounds For Rejection To Be Argued On Appeal:

I.   Claims 4-5, 8, 23-24, and 27 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite. More specifically, the Examiner states that claims 4, 5, 8, 23, 24, and 27 fail to particularly point out and distinctly claim the subject matter which Applicants regard as their invention because those claims contain the trademark/trade name Java.

II.   Claims 1-3, 6-7, 20-22, 25-26, 33, 36, and 38 stand rejected under 35 U.S.C. §102(e) as being fully anticipated by U.S. Patent No. 6,292,936 to *Wang* (hereinafter "Wang").

III.   Claims 4-5, 8, 23-24, and 27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Wang in view of U.S. Patent No. 6,732,330 to *Claussen* (hereinafter "Claussen").

IV.   Claims 9-13, 28-32, 34-35, and 37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Wang in view of U.S. Patent No. 6,732,330 to *Connor et al.* (hereinafter "Connor").

Arguments:

   I.   <u>Rejections of claims 4-5, 8, 23-24, and 27 under 35 U.S.C. §112, second paragraph, were improper because Applicants' use of the trademark Java in those claims does not render the claims indefinite.</u>

According to MPEP 608.01(v), ". . . the use of trademarks having definite meanings is permissible in patent applications . . . ." Applicants submit that Java has a definite meaning within the art. Java, in fact, is one of the two predominant object-oriented programming languages, and has been used in countless software applications. The definite meaning of "Java" has been consistently recognized by the United States Patent and Trademark Office. A search of its "USPTO Patent Full-text and Image Database" using "Java" as the keyword, and looking only in the claims of issued patents, reveals 1,240 patents having the mark "Java" at least once in their claims (search performed Jan. 30, 2006). Example patents include: U.S. Patent No.

- 10 -

6,938,214 to *Proulx, et al.*; U.S. Patent No. 6,938,041 to *Brandow, et al.*; and U.S. Patent No. 6,754,884 to *Lucas, et al.* Thus, it is apparent that "Java" does in fact have a definite meaning, and under MPEP 608.01(v), can be used in the claims and specification of the instant application.

It is accordingly believed that the specification and the claims meet the requirements of 35 U.S.C. § 112, second paragraph.

II. <u>Rejections of claims 1-3, 6-7, 20-22, 25-26, 33, 36, and 38 under 35 U.S.C. §102(e) were improper because Wang fails to anticipate the claimed invention as claimed in claims 1-3, 6-7, 20-22, 25-26, 33, 36, and 38.</u>

It is well settled that anticipation under 35 U.S.C. §102 requires the disclosure in a single piece of prior art to teach **each and every** limitation of a claimed invention. *Electro Med. Sys. S.A. v. Cooper Life Sciences*, 34 F.3d 1048, 1052, 32 USP Q2d 1017, 1019 (Fed. Cir. 1994). . MPEP 2131 states, "TO ANTICIPATE A CLAIM, THE REFERENCE MUST TEACH EVERY ELEMENT OF THE CLAIM" and "a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Furthermore, anticipation requires that each claim element must be <u>identical</u> to a corresponding element in the applied reference. *Glaverbel Société Anonyme v. Northlake Mktg & Supply, Inc.*, 45 F.3d 1550, 1554 (Fed. Cir. 1995). Thus, to anticipate the present invention, Wang must disclose every element recited in the pending claims.

Claims 1-3, 6-7, 20-22, and 25-26
Claim 1 calls for a "method of computing comprising:

SEA/109870/130097/RPE/358785.1

reading a data processing representation having code sections with code

statements of at least a first and a second programming language;

recognizing a first code section with at least code statements of a first

programming language;

invoking a first code statement processing unit of the first programming language

to process the first code section;

recognizing a second code section with at least code statements of a second

programming language; and

invoking a second code statement processing unit of the second programming

language to process the second code section."

Accordingly, claim 1 is directed towards a multilanguage processing method, with first and second processing units of a first programming language and a second programming language, being respecitvely invoked to process first and second code sections of the first and second programming languages that have been recognized in a data processing representation being read.

In contrast, Wang merely teaches "an interpreter-based scripting environment [that] includes multiple runtime processors executed by the computer. Each of the runtime processors processes their respective corresponding intermediate sources derived from an original source in a synchronous manner" (abstract). Specifically, the original source disclosed in Wang comprises an HTML document with embedded Visual Basic scripting langauge blocks. Wang teaches an HTML parser that parses the original HTML and VB source, and translates the non-VB source into a first intermediate source executable by a Java VM, and the VB source into a second intermediate source having the VB script statements executable by a VB script interpreter.

Thus, under Wang a SINGLE parser (the HTML parser is employed to "process" (translate) the two types of code sections encountered (the HTML and the VB sources), to translate the two types of code into two corresponding types of intermediate sources.

Therefore, the translation teaching of Wang does not anticipate the required recitations of claim 1.

Under Wang, two separate processing units (Java VM and VB script interpreter) are eventually invoked to process the two types of intermediate sources. The two separate processing units (Java VM and VB script interpreter) cannot be said to process the two code sections of the data representation being read, as required by the recitations of claim 1, since the two separate processing units (Java VM and VB script interpreter) process derived intermediate codes of the two code sections of the data representation being read, <u>not</u> the two code sections of the data representation being read themselves.

Thus, for the foregoing reasons, Wang does not teach or anticipate at least the two required "invoking" recitations of claim 1. Accordingly, claim 1 is patentable over Wang under §102(e).

Claim 20 includes similar language directed to an apparatus of the invention. Thus, for at least the same reasons, claim 20 is patentable over Wang.

Claims 2, 3, 6, 7, 21, 22, 25, and 26 depend from claims 1 and 20, incorporating their limitations respectively. Thus, for at least the same reasons, claims 3, 6, 7, 21, 22, 25, and 26 are patentable over Wang.

Additionally, even assuming Wang anticipates claims 1 and 6, and/or claims 20 and 25, Wang does not disclose, expressly or inherently, that "said third code section is embedded within said second code section, and said second code section is embedded within said first code section," as is claimed in claims 7 and 26. While Wang does teach an original input source, such as HTML, that may include a plurality of embedded code sections, such as embedded VB Script, Wang does not disclose a third code section that may be embedded in a second code section when the second is itself embedded in a first. The example given by Wang of a second code section (VB Script) embedded in

- 13 -

a first code section (arguably, HTML) does make reference to the possibility of other additional code sections, but makes no explicit reference to such sections being embedded in VB Script or any other code section that might itself be embedded in HTML. Accordingly, Wang does not anticipate claims 7 and 26.

Claims 33, 36, and 38

Claim 33 calls for an "apparatus comprising:

at least one storage medium having stored therein a plurality of programming instructions designed to enable the apparatus to

read a data processing representation having code sections with programming language statements of at least a first and a second programming language,

recognize a header section of a selected one of the first and the second programming language;

recognizing a directive statement within the header section, enumerating one or more data packages, and

import the enumerated one or more data packages for use code sections with at least code statements of the selected one of the first and the second programming language; and

at least one processor coupled to the storage medium to execute the programming instructions."

Although Wang arguably makes reference to reading "a data processing representation having code sections with programming language statements of at least a first and a second programming language," no explicit mention is made of recognizing a header section in at least one of the representation's programming language code sections. Nor is any "recognizing" operation inherent in Wang. The method and scripting environment of Wang merely requires that an HTML parser recognize code sections of different languages. No other recognition operations are required.

Accordingly, claim 33 is patentable over Wang under §102(e).

Claims 36 and 38 include similar language directed to other apparatuses of the invention. Thus, for at least the same reasons, claims 36 and 38 are patentable over Wang.

III.   Rejections of claims 4-5, 8, 23-24, and 27 under 35 U.S.C. §103(a) were improper because Wang and Claussen, alone or in combination, fail to teach or suggest the claimed invention when the invention as claimed in claims 4-5, 8, 23-24, and 27 is viewed as a whole.

As stated above, Wang fails to teach the required, recited operations of the present invention, as claimed in claims 1 and 20. Claussen does not remedy the above discussed deficiencies of Wang. Thus, even when combined with Claussen, the cited art fails to disclose or suggest the novel features that are noted when the invention of claims 1 and 20 is viewed as a whole.

Claims 4-5, 8, 23-24, and 27 depend from claims 1 and 20, respectively. Consequently, claims 4-5, 8, 23-24, and 27 are patentable over the combination of Wang and Claussen, under 35 U.S.C. §103(a).

Additionally, even assuming for the sake of argument some teaching to combine Wang and Claussen (Applicants do not concede that any such teaching exists), Claussen fails to disclose "wherein said first language is a directive language, and said second language is a selected one of XML and Java," as recited by claim 4 (claims 5, 8, 23, 24, and 27 contain similar recitations to the above mentioned recitation of claim 4). Rather, Claussen teaches "a web page . . . supporting multiple scripting languages is compiled into an XML . . . DOM (Document Object Model), and, thereafter, into a Java servlet." While multiple scripting languages are mentioned, Claussen simply does not suggest the use of a directive language, XML, or Java as one or more of the multiple

- 15 -

scripting languages. Instead, Claussen teaches the compiling of said languages into XML, and then into a Java servlet. Therefore, Claussen does not teach the limitation lacking in Wang, and thus when combined with Wang cannot form the basis for an obviousness rejection.

IV.   Rejections of claims 9-13, 28-32, 34-35, and 37 under 35 U.S.C. §103(a) were improper because Wang and Connor, alone or in combination, fail to teach or suggest the claimed invention when the invention as claimed in claims 9-13, 28-32, 34-35, and 37 is viewed as a whole.

As stated above, Wang fails to teach the required, recited operations of the present invention, as claimed in claims 1, 20, 33, and 36. Connor does not remedy the above discussed deficiencies of Wang. Thus, even when combined with Connor, the cited art fails to disclose or suggest the novel features that are noted when the invention of claims 1, 20, 33, and 36 is viewed as a whole.

Claims 9-13, 28-32, 34-35, and 37 depend from claims 1, 20, 33, and 36, respectively. Consequently, claims 9-13, 28-32, 34-35, and 37 are patentable over the combination of Wang and Connor, under 35 U.S.C. §103(a).

Conclusion

Appellants respectfully submit that all the appealed claims in this application are patentable and requests that the Board of Patent Appeals and Interferences overrule the Examiner and direct allowance of the rejected claims.

This brief is submitted with Check Number 13968 for $500.00 to cover the filing of the appeal brief. We do not believe any additional fees, in particular extension of time fees, are needed. However, should that be necessary, please charge our deposit account 500393. In addition, please charge any shortages and credit any overages to Deposit Account No. 500393.

Respectfully submitted,

Date: June 12, 2006
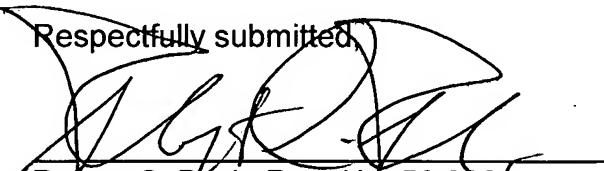
Robert C. Peck, Reg. No. 56,826
Agent for Appellant Applicant

Schwabe Williamson & Wyatt, P.C.
1420 Fifth, Suite 3010
Seattle, WA 98101
Tel: (206) 622-1711
Fax: (206) 292-0460

Appendix A – Appealed Claims

1.   (Original) A method of computing comprising:

reading a data processing representation having code sections with code
statements of at least a first and a second programming language;

recognizing a first code section with at least code statements of a first
programming language;

invoking a first code statement processing unit of the first programming
language to process the first code section;

recognizing a second code section with at least code statements of a second
programming language;

invoking a second code statement processing unit of the second programming
language to process the second code section.

2.   (Previously Presented) The method of claim 1, wherein the first and second
code sections are non-interleaved code sections.

3.   (Original) The method of claim 1, wherein said second code section is
embedded within said first code section.

4.   (Original) The method of claim 1, wherein said first language is a directive
language, and said second language is a selected one of XML and Java.

5.   (Original) The method of claim 1, wherein said first language is Java, and said
second language is XML.

6.   (Previously Presented) The method of claim 1, wherein the method further
comprises

recognizing a third code section with at least code statements of a third
programming language; and

SEA/109870/130097/RPE/358785.1

invoking a third code statement processing unit of the third programming
language to process the third code section.

7.   (Original) The method of claim 6, wherein said third code section is embedded
within said second code section, and said second code section is embedded within
said first code section.

8.   (Original) The method of claim 6, wherein said first language is a directive
language, said second language is Java and said third language is XML.

9.   (Previously Presented) The method of claim 1, wherein the method further
comprises
        recognizing an invocation of a library function within at least a selected one of
                said first and second code sections; and
        invoking the library function, and outputting the result of the invocation.

10.   (Original) The method of claim 9, wherein the library function is a selected one
of an emit function for outputting execution results, a pop function for returning an
element, and a push function for backing up an insertion point.

11.   (Original) The method of claim 1, wherein the method further comprises
        recognizing a header section of a selected one of the first and the second
                programming language;
        recognizing a directive statement within the header section, enumerating one or
                more data packages; and
        importing the enumerated one or more data packages for use within code
                sections with at least statements of the selected first and second
                programming language.

12.   (Original) The method of claim 1, wherein the method further comprises

recognizing a header section of a selected one of the first and the second programming language;

recognizing a declare statement within the header section, enumerating one or more processing methods; and

instantiating the enumerated one or more processing methods for use within code sections with at least statements of the selected first and second programming language.

13.   (Original) The method of claim 1, wherein the method further comprises recognizing a header section of a selected one of the first and the second programming language;

recognizing a declare statement within the header section, enumerating one or more instance variables; and

instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language.

14.   (Original) A method of computing comprising:

reading a data processing representation having code sections with code statements of at least a first and a second programming language;

recognizing a header section of a selected one of the first and the second programming language;

recognizing a directive statement within the header section, enumerating one or more data packages; and

importing the enumerated one or more data packages for use by code sections within code sections with at least statements of the selected first and second programming language.

15.   (Original) The method of claim 14, wherein the method further comprises recognizing a declare statement within the header section, enumerating one or more processing methods; and

instantiating the enumerated one or more processing methods for use within
code sections with at least statements of the selected first and second
programming language.

16.   (Original) The method of claim 14, wherein the method further comprises
recognizing a declare statement within the header section, enumerating one or
more instance variables; and
instantiating the enumerated one or more instance variables for use within code
sections with at least statements of the selected first and second
programming language.

17.   (Original) A method of computing comprising:
reading a data processing representation having code sections with code
statements of at least a first and a second programming language;
recognizing a header section of a selected one of the first and the second
programming language;
recognizing a first declare statement within the header section, enumerating
one or more processing methods; and
instantiating the enumerated one or more processing methods for use within
code sections with at least statements of the selected first and second
programming language.

18.   (Original) The method of claim 17, wherein the method further comprises
recognizing a second declare statement within the header section, enumerating
one or more instance variables; and
instantiating the enumerated one or more instance variables for use within code
sections with at least statements of the selected first and second
programming language.

19.   (Original) A method of computing comprising:

reading a data processing representation having code sections with code statements of at least a first and a second programming language;

recognizing a header section of a selected one of the first and the second programming language;

recognizing a declare statement within the header section, enumerating one or more instance variables; and

instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language.

20.   (Original) An apparatus comprising:

at least one storage unit having stored thereon programming instructions designed to enable the apparatus to

read a data processing representation having code sections with code statements of at least a first and a second programming language,

recognize a first code section with code statements of at least the first programming language,

invoking a first code statement processing unit of the first programming language to process the first code section,

recognize a second code section with code statements of at least the second programming language,

invoking a second code statement processing unit of the second programming language to process the second code section; and

at least one processor coupled to said at least one storage unit to execute said programming instructions.

21.   (Previously Presented) The apparatus of claim 20, wherein the first and second code sections are non-interleaved code sections.

22.   (Original) The apparatus of claim 20, wherein said second code section is embedded within said first code section.

23.   (Original) The apparatus of claim 20, wherein said first language is a directive language, and said second language is a selected one of XML and Java.

24.   (Original) The apparatus of claim 20, wherein said first language is Java, and said second language is XML.

25.   (Previously Presented) The apparatus of claim 20, wherein the programming instructions further enable the apparatus to
        recognize a third code section with at least code statements of a third
                programming language; and
        invoke a third code statement processing unit of the third programming
                language to process the third code section.

26.   (Original) The apparatus of claim 25, wherein said third code section is embedded within said second code section, and said second code section is embedded within said first code section.

27.   (Original) The apparatus of claim 25, wherein said first language is a directive language, said second language is Java and said third language is XML.

28.   (Previously Presented) The apparatus of claim 20, wherein said programming instructions further enable the apparatus to
        recognize an invocation of a library function of a selected one of the first and
                the second programming language within the first code section; and
        invoke the library function, and output the result of the invocation.

29.   (Original) The apparatus of claim 28, wherein the library function is a selected one of an emit function for outputting execution results, a pop function for returning an element, and a push function for backing up an insertion point.

30.   (Original) The apparatus of claim 20, wherein the said programming instructions are further designed to enable the apparatus to

    recognize a header section of a selected one of the first and the second programming language;

    recognize a directive statement within the header section, enumerating one or more data packages; and

    import the enumerated one or more data packages for use by code sections with at least code statements of the selected one of the first and the second programming language.

31.   (Original) The apparatus of claim 20, wherein said programming instructions are further designed to enable the apparatus to

    recognize a header section of a selected one of the first and the second programming language;

    recognize a declare statement within the header section, enumerating one or more processing methods; and

    instantiate the enumerated one or more processing methods for use within code sections with at least code statements of the selected one of the first and the second programming language.

32.   (Original) The apparatus of claim 20, wherein said programming instructions are further designed to enable the apparatus to

    recognize a header section of a selected one of the first and the second programming language;

    recognize a declare statement within the header section, enumerating one or more instance variables; and

    instantiate the enumerated one or more instance variables for use code sections with at least code statements of the selected one of the first and the second programming language.

33.   (Original) An apparatus comprising:

at least one storage medium having stored therein a plurality of programming
instructions designed to enable the apparatus to

read a data processing representation having code sections with programming
language statements of at least a first and a second programming
language,

recognize a header section of a selected one of the first and the second
programming language;

recognizing a directive statement within the header section, enumerating one or
more data packages, and

import the enumerated one or more data packages for use code sections with
at least code statements of the selected one of the first and the second
programming language; and

at least one processor coupled to the storage medium to execute the
programming instructions.

34.   (Original) The apparatus of claim 33, wherein said programming instructions
are further designed to enable the apparatus to

recognize a declare statement within the header section, enumerating one or
more processing methods, and

instantiate the enumerated one or more processing methods for use within code
sections with at least code statements of the selected one of the first and
the second programming language.

35.   (Original) The apparatus of claim 33, wherein said programming instructions
are further designed to enable the apparatus to

recognize a declare statement within the header section, enumerating one or
more instance variables, and

instantiate the enumerated one or more instance variables for use within code
sections with at least code statements of the selected one of the first and
the second programming language.

36. (Original) An apparatus comprising:

at least one storage medium having stored therein a plurality of programming instructions designed to enable the apparatus to

read a data processing representation having code sections with code statements of at least a first and a second programming language,

recognize a header section of a selected one of the first and the second programming language,

recognize a first declare statement within the header section, enumerating one or more processing methods, and

instantiate the enumerated one or more processing methods for use within code sections with at least code statements of the selected one of the first and the second programming language; and

at least one processor coupled to the storage medium to execute the programming instructions.

37. (Original) The apparatus of claim 36, wherein said programming instructions are further designed to enable the apparatus to

recognize a second declare statement within the header section, enumerating one or more instance variables, and

instantiate the enumerated one or more instance variables for use within code sections with at least code statements of the selected one of the first and the second programming language.

38. (Original) An apparatus comprising:

at least one storage medium having stored therein a plurality of programming instructions designed to enable the apparatus to

read a data processing representation having code sections with code statements of at least a first and a second programming language,

recognize a header section of a selected one of the first and the second programming language,

SEA/109870/130097/RPE/358785.1

recognize a declare statement within the header section, enumerating one or more instance variables,

instantiate the enumerated one or more instance variables for use within code sections with at least code statements of the selected one of the first and the second programming language; and

at least one processor coupled to the storage medium to execute the programming instructions.

Appendix B – Copies of Evidence Submitted

No evidence has been submitted under 37 C.F.R. 1.130, 1.131, or 1.132. No evidence entered by Examiner has been relied upon by Appellants in the appeal.

Appendix C – Related Proceedings

There are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.
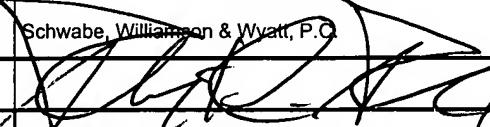
TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/089,139
Filing Date	March 22, 2002
First Named Inventor	Adam Bosworth
Art Unit	2191
Examiner Name	Rampuria, Satish
Attorney Docket Number	109870-130097

Total Number of Pages in This Submission

ENCLOSURES *(Check all that apply)*

- [✓] Fee Transmittal Form
  - [✓] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53

- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD

- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [✓] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [✓] Other Enclosure(s) (please Identify below):
  Return Receipt Postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Schwabe, Williamson & Wyatt, P.C.
Signature	
Printed name	Robert C. Peck
Date	June 12, 2006	Reg. No.	56,826

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	
Typed or printed name	Yvette L. Chriscaden	Date	June 12, 2006

Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2006

Complete if Known	
Application Number	10/089,139
Filing Date	March 22, 2002
First Named Inventor	Adam Bosworth
Examiner Name	Rampuria, Satish
Art Unit	2191
Attorney Docket No.	109870-130097

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($)	500

METHOD OF PAYMENT (check all that apply)

☑ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☑ Deposit Account Deposit Account Number: 500393   Deposit Account Name: Schwabe, Williamson et al

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☐ Charge fee(s) indicated below   ☐ Charge fee(s) indicated below, **except for the filing fee**

☑ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17   ☑ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION (All the fees below are due upon filing or may be subject to a surcharge.)

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

Application Type	FILING FEES Fee ($)	Small Entity Fee ($)	SEARCH FEES Fee ($)	Small Entity Fee ($)	EXAMINATION FEES Fee ($)	Small Entity Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	_____
Design	200	100	100	50	130	65	_____
Plant	200	100	300	150	160	80	_____
Reissue	300	150	500	250	600	300	_____
Provisional	200	100	0	0	0	0	_____

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 20 or HP =	_____ x _____ = _____		

HP = highest number of total claims paid for, if greater than 20.

Indep. Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 3 or HP =	_____ x _____ = _____		

HP = highest number of independent claims paid for, if greater than 3.

Multiple Dependent Claims

Fee ($)	Fee Paid ($)
_____	_____

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee ($)	Fee Paid ($)
_____ - 100 =	_____ / 50 =	_____ (round up to a whole number) x	_____ =	_____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	_____
Other (e.g., late filing surcharge): Appeal Brief Filing Fee	500

SUBMITTED BY

Signature	[signature]	Registration No. (Attorney/Agent) 56,826	Telephone 503-222-9981
Name (Print/Type) Robert C. Peck			Date June 12, 2006